CLAIMS

what is claimed is:

1. A method of producing a simulation model of an electronic design, which simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the electronic design, the method comprising:

receiving a non-obfuscated version of the electronic design suitable for direct compilation to a practical hardware implementation of the electronic design; and

adding obfuscation circuitry to produce an obfuscated version of the electronic design from which the simulation model can be created,

wherein the obfuscation circuitry does not substantially impact the accuracy of the simulation result, but prevents practical implementation of the electronic design on a hardware device.

2. A method as recited in claim 1, wherein the non-obfuscated version of the electronic design is provided in a HDL source format.

3. A method as recited in claim 1, wherein the electronic design is a reusable IP core.

4. A method as recited in claim 1, wherein adding obfuscation circuitry comprises:

identifying a region for introduction of obfuscation circuitry in the non-obfuscated version of the electronic design;

choosing a type of obfuscation circuitry for insertion; and

inserting the chosen type of obfuscation circuitry into the identified region, thereby creating an obfuscated region.

5. A method as recited in claim 4, wherein identifying a region for introduction of obfuscation circuitry comprises identifying in the non-obfuscated version of the electronic design logic of a type that is not removed by a synthesizer.

6. A method as recited in claim 5, wherein the type of logic that is not removed by a synthesizer comprises one or more flip-flops.

7. A method as recited in claim 1, further comprising optimizing the obfuscated version of the electronic design to merge the obfuscation circuitry with functional circuitry.

8. A method as recited in claim 1, wherein the obfuscation circuitry comprises circuitry that increases the size of the electronic design without changing its function and/or slows the speed of the electronic design without changing its function.

9. A method as recited in claim 1, wherein adding obfuscation circuitry comprises:

at a first location, adding circuitry for scrambling an input signal by spreading out the input signal in time; and

at a second location, adding circuitry for de-scrambling an output signal resulting from the circuitry for scrambling.

10. A method as recited in claim 1, wherein adding obfuscation circuitry comprises:

at a first location, adding circuitry for entangling multiple input signals to thereby spread out the input signals; and

at a second location, adding circuitry for detangling an output signal resulting from the circuitry for entangling.

11. A method as recited in claim 1, wherein the obfuscation circuitry comprises a XOR tree.

12. A method as recited in claim 1, wherein adding obfuscation circuitry is performed automatically.

13. An apparatus for producing a simulation model of an electronic design, which simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the electronic design, the apparatus comprising:

one or more processors;

memory; and

an obfuscation module for adding obfuscation circuitry to a non-obfuscated version of the electronic design to produce an obfuscated version of the electronic design from which the simulation model can be created, wherein the obfuscation circuitry does not substantially impact the accuracy of the simulation result, but prevents practical implementation of the electronic design on a hardware device.

14. An apparatus as recited in claim 13, wherein the non-obfuscated version of the electronic design is in a HDL source format.

15. An apparatus as recited in claim 13, wherein the electronic design is a reusable IP core.

16. An apparatus as recited in claim 13, wherein the obfuscation module comprises:

a scanning module for identifying a region for introduction of obfuscation circuitry in the non-obfuscated version of the electronic design;

a selection module for choosing a type of obfuscation circuitry for insertion; and

an insertion module for inserting the chosen type of obfuscation circuitry into the identified region, thereby creating an obfuscated region.

17. An apparatus as recited in claim 16, wherein the scanning module for identifying a region for introduction of obfuscation circuitry comprises identifying in the non-obfuscated version of the electronic design logic of a type that is not removed by a synthesizer.

18. An apparatus as recited in claim 17, wherein the type of logic that is not removed by a synthesizer comprises one or more flip-flops.

19. An apparatus as recited in claim 13, further comprising:

an optimizer for optimizing the obfuscated version of the electronic design to merge the obfuscation circuitry with functional circuitry.

20. An apparatus as recited in claim 13, wherein the obfuscation circuitry comprises circuitry that increases the size of the electronic design without changing its function and/or slows the speed of the electronic design without changing its function.

21. An apparatus as recited in claim 13, wherein the obfuscation circuitry comprises of:

at a first location, a scrambler having circuitry for scrambling an input signal by spreading out the input signal in time; and

at a second location, a descrambler having circuitry for de-scrambling an output signal resulting from the circuitry for scrambling.

22. An apparatus as recited in claim 13, wherein the obfuscation circuitry comprises of:

at a first location, an entangler having circuitry for entangling multiple input signals to thereby spread out the input signals; and

at a second location, a detangler having circuitry for detangling an output signal resulting from the circuitry for entangling.

23. An apparatus as recited in claim 13, wherein the obfuscation circuitry comprises a XOR tree.

24. An apparatus as recited in claim 16, wherein the scanning module, the selection module, and the insertion module are configured to operate automatically.

25. A computer program product comprising a machine readable medium on which is provided program instructions for producing a simulation model of an electronic design, which simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the electronic design, the program instructions comprising:

instructions for receiving a non-obfuscated version of the electronic design suitable for direct compilation to a practical hardware implementation of the electronic design; and

instructions for adding obfuscation circuitry to produce an obfuscated version of the electronic design from which the simulation model can be created,

wherein the obfuscation circuitry does not substantially impact the accuracy of the simulation result, but prevents practical implementation of the electronic design on a hardware device.

26. A computer program product as recited in claim 25, wherein the non-obfuscated version of the electronic design is provided in a HDL source format.

27. A computer program product as recited in claim 25, wherein the electronic design is a reusable IP core.

28. A computer program product as recited in claim 25, wherein the instructions for adding obfuscation circuitry comprises:

instructions for identifying a region for introduction of obfuscation circuitry in the non-obfuscated version of the electronic design;

instructions for choosing a type of obfuscation circuitry for insertion; and

instructions for inserting the chosen type of obfuscation circuitry into the identified region, thereby creating an obfuscated region.

29. A computer program product as recited in claim 28, wherein the instructions for identifying a region for introduction of obfuscation circuitry comprises identifying in the non-obfuscated version of the electronic design logic of a type that is not removed by a synthesizer.

30. A computer program product as recited in claim 29, wherein the type of logic that is not removed by a synthesizer comprises one or more flip-flops.

31. A computer program product as recited in claim 28, further comprising:

instructions for optimizing the obfuscated version of the electronic design to merge the obfuscation circuitry with functional circuitry.

32. A computer program product as recited in claim 25, wherein the obfuscation circuitry comprises circuitry that increases the size of the electronic design without changing its function and/or slows the speed of the electronic design without changing its function.

33. A computer program product as recited in claim 25, wherein the instructions for adding obfuscation circuitry comprises:

instructions for adding circuitry at a first location to scramble an input signal by spreading out the input signal in time; and

instructions for adding circuitry at a second location to de-scrambling an output signal resulting from the circuitry to scramble.

34. A computer program product as recited in claim 25, wherein the instructions for adding obfuscation circuitry comprises:

instructions for adding circuitry at a first location to entangle multiple input signals to thereby spread out the input signals; and

instructions for adding circuitry at a second location to detangle an output signal resulting from the circuitry to entangle.

35. A computer program product as recited in claim 25, wherein the obfuscation circuitry comprises a XOR tree.

36. A computer program product as recited in claim 28, wherein the operations of identifying, choosing, and inserting can be done automatically.

37. A method of producing a simulation model of an IP core, wherein the simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the IP core, the method comprising:

(a) receiving a non-obfuscated version of the IP core in a native HDL format or in a partially compiled HDL format;

(b) identifying a region of the non-obfuscated IP core where one or more flip-flops are located;

(c) inserting entangler circuitry upstream from the region and inserting complementary detangler circuitry downstream from the region;

(d) inserting scrambler circuitry upstream from the region and inserting complementary descrambler circuitry downstream from the region; and

(e) optimizing the IP core after the insertions of (c) and (d).

38. A method of producing a simulation model of an IP core, wherein the simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the IP core, the method comprising:

(a) receiving a non-obfuscated version of the IP core in a native HDL format or in a partially compiled HDL format;

(b) identifying a region of the non-obfuscated IP core where one or more flip-flops are located;

(c) inserting obfuscation circuitry into the region;

(d) adding additional flip-flops and/or modifying the flip-flops; and

(e) optimizing the IP core after (c) and (d) have been performed.

39. An IP core comprising:

a programming version of the IP core for insertion in an electronic design developed using a specified EDA platform; and

a simulation model of the IP core for simulating operation of the IP core in the electronic design, wherein the simulation model comprises obfuscation circuitry, absent in the programming version, which allows a hardware simulation result of the IP core but prevents direct compilation to produce a practical hardware implementation of the IP core.